Sentiment classification with GST tweet data on LSTM based on polarity-popularity model

SOURAV DAS[1,*], DIPANKAR DAS[2] and ANUP KUMAR KOLYA[3]

[1]National Institute of Electronics & Information Technology (NIELIT), Jadavpur, Kolkata, India
[2]Department of Computer Science & Engineering, Jadavpur University, Jadavpur, Kolkata, India
[3]Department of Computer Science & Engineering, RCC Institute of Information Technology, Beleghata, Kolkata, India
e-mail: sourav.das17.91@gmail.com; dipankar.dipnil2005@gmail.com; anup.kolya@gmail.com

Abstract. One of the biggest issues of Indian economy in 2017 was the implementation of Goods and Services Tax (GST), and the social networks witnessed a lot of opinion contrasts and conflicts regarding this new taxation system. Inspired by such a large-scale tax reformation, we developed an experimental approach to analyze the reactions of public sentiment on Twitter based on popular words either directly or indirectly related to GST. We collected a number of almost 200 k tweets solely about GST from June 2017 to December 2017 in two phases. In order to assure the relevance of our crawled tweets with respect to GST, we prepared a topic-sentiment relevance model. Furthermore, we employed several state-of-the-art lexicons for identifying sentiment words and assigned polarity ratings to each of the tweets. On the other hand, in order to extract the relevant words that are linked with GST implicitly, we propose a new polarity-popularity framework and such popular words were also rated with sentiments. Next, we trained an LSTM model using both types of rated words for predicting sentiment on GST tweets and obtained an overall accuracy of 84.51%. It was observed that the performance of the system has been started improving while incorporating the knowledge of indirectly related GST words during training.

Keywords. Deep learning; GST; lexicons; LSTM; sentiment analysis; Twitter.

1. Introduction

Twitter-based research on sentiment analysis is being popular as much as the micro-blogging platforms are becoming the open stages to the public for expressing their views in short texts and/or emotions. Twitter sentiment analysis not only helps to identify the mass opinion on certain topics, but it also captures the ongoing as well as future trends of the social dynamics, political scenario, and even economy of a country. Unlike Facebook, Twitter users cannot form any group on the basis of a similar perspective or ideology. Rather, on Twitter, people with a similar type of choice tend to follow the same people or the Twitter handle of similar types of organizations.

In recent trends, Twitter witnessed a large number of events or movements like the 2016 USA's presidential election[1] or anti-harassment movement like Me Too[2] or demonetization[3] as well as the implementation of Goods & Service Tax (GST)[4] in India. GST can be stated as one of the largest tax reforms for the achievement of "one nation one tax" system in the post-independence history of India.

People's curiosity and opinion about GST attained its peak when it was implemented and its importance highly motivated us to gain the actual insights of this new taxation system for the world's largest democracy.

On the other hand, one of the important challenges of collecting tweets from Indian users in any context is due to multilingualism. Often the native speakers use more than one language while posting their tweets in order to avail comfort in communication. This property of mixing more than one language is treated as code-mixing [1]. Moreover, the tweet users often use more than one script also. Thus, the handling of code-mixed as well as cross script data is becoming an important research challenge to deal with in Indian context.

Here, we present a topic-sentiment model inspired by popularity and polarity of words to analyze the public sentiments on GST in Twitter over time. Mostly, GST tweets of Indian users were collected in two phases over the

---

[1]www.usnews.com.

[2]http://time.com.

[3]https://www.rbi.org.in.

[4]www.economictimes.indiatimes.com.

duration of seven months from its implementation in June-July 2017 to the reformation of GST by the GST council in its 23rd meeting on November 10, 2017, till the later part of the same month. However, for our approach, we only considered pure English tweets. We developed a Twitter dataset consisting of almost 200 k or 2 lakhs English tweets solely on GST issue using Twitter streaming API and refined the tweets strictly to their relevance with GST using the topic-sentiment model. Employing various state-of-the-art sentiment lexicons as comparative parameters and Naïve Bayes Bag of Words (BOW) model to our labeled data corpus, the system identifies the sentiment rated word clusters as well as polarity assigned tweets related to GST. Therefore, in order to track the trend with more GST related issues; a polarity-popularity model has been implemented. We further applied it to preserve each of the topic words exclusively for an event like this with a motivation to engage them in identifying similar future events to predict the scenario and effect of it upon the economy or even on the society. With that, we have also demonstrated the polarity mapping or relation of such words within a given miniature range of tweets. These words from the clusters carry respective probability scores, i.e., they reflect the dense or sparse possibilities to appear within the tweets about GST.

These labelled tweets along with polarity-popularity rated words were fed into our deep LSTM model for its training and testing on GST twitter data [2]. Since it is already shown that long-short term memory models are more effective almost every time for accomplishing the task on sentiment analysis [3], we also used the multi-layer LSTM model with multi-activation functions for achieving better accuracy and representing tweet trends over the course of seven months of our data collection. Our approach combines an input of two feature sets phase by phase. We developed the LSTM model capable of taking two distinct input sets as Feature I and Feature II. In Feature I, we did n-gram extraction from our GST data corpus and compared them with some previous gold standard datasets. Then, we determined GST exclusive word generation using polarity-popularity model. Feature II deals with the sentiment rating of GST tweets.

The organization of the rest of the paper is described as follows. In section 2, we discuss briefly some relevant literature review. In section 3, we present a complete insight of our dataset and stages of pre-processing. In section 4, we discussed tweet models consists of topic and sentiment models. Section 5 describes the approach of identifying GST-specified sentiment words using state-of-the-art lexicons as the first feature set whereas section 6 mentions the new polarity-popularity model and its application for extracting GST-implied words that are indirectly related to identifying sentiments from GST tweets. The sentiment rating process is discussed in section 7 as the second feature set followed by section 8 where we present our LSTM model as for sentiment prediction using the 80:20 split data validation. Section 9 highlights the results, error cases and discussions, which focus further comparison with the accuracy predicted by the LSTM model. Finally, in section 10, we conclude and pave the future prospects of our research.

2. Related work

Earlier studies show that the named entity classifier problem has already been demonstrated for both single objective and multi-objective ensemble approach [4]. With the strength of predictions and outputs of each classifier tends to differ from class to classes, hence it is necessary to find out the better class within an ensemble system to find out the better outcomes and predictions. The researchers here used a number of seven distinct classifiers to build several heterogeneous models like a black-box tool, without using any supervised or prior language-specific library knowledge. They primarily implied the model for less-resourced regional Indian languages such as Bengali, Hindi, and Telugu, while the multi-objective optimization-based approach proposed by the authors claimed to be the most successful among the model.

As tweets can be expressed as instant dynamic textual segments, hence one main problem with tweets is, in general, they are unstructured and noisy. Tweets can contain a lot of misspelled words, unnecessary punctuation marks, and several other impurities. In a shared task work from 2013, have addressed this issue regarding noisy twitter data, where natural English parsers or POS tagging does not perform as per expectations [5]. Authors here proposed a model to detect polarity from discourse relations. Alongside they also proposed how the inherent conjunctions, connectives, modals, and conditionals affect the polarity construction within tweets. Also, tweets can commonly contain abbreviations, popular SMS terminologies, slangs, etc., which the authors have also taken into account.

A popular domain of work in NLP is to develop a machine translation bridge to perform sentiment analysis of a less recourse language from an annotated more resourceful language [6]. It can be done if the latter's developed corpora; POS tagged, stemming or lemmatized is readily available and already well-rehearsed. However, the success of this concept also depends on the availability of the machine translation system between the two languages.

Another research work shows a system for real-time twitter data (or tweets) analysis of the US presidential election [7]. This was an event-based sentiment analysis work, which relies heavily on time and content. With that, the researchers wanted to portray the aggregation and visualization of their key found results. Since tweets are dynamic and expressed with a short span, hence users also tend to tweet sarcastically, crack short jokes, in humorous

or satirical ways. Another previous work shows any of the past sarcastic tweets about any topic of the authors match with their present-day ideology or not [8]. For this, the authors have started to use a bi-predictor approach to determine the sentiment contrast for sensing the sarcasm within a large tweet, along with another predictor to find out the historical sarcastic tweet of the same user on any given topic, if any. Also, their approach is said to gather the texts generated by the author while tweeting, to detect the sarcasm within them.

One of the primary problems of working on tweets is that when the tweets are newly collected from twitter, they often consist of misspelled words, wrong and exaggerating the use of punctuation marks, unnecessary emojis or emoticons and so on. These impure tweets can be said as noisy tweets or noisy data from a broader perspective. Noisy data is not appropriate for labeling or processing; hence tweet filtration is a must need just after collecting a tweet corpus. A group of researchers have addressed this issue in their work. They have worked for the normalization of the noisy tweets based on their lexical and syntactic properties using a hybrid approach [9]. This approach consists of a combination of the machine learning algorithm and rule-based classifier. The machine learning algorithm used is a supervised "conditional random field", which is first developed. Henceforth in the second step, a set of heuristics rules has been applied to the word forms gained from the first step, for normalizing them. The researchers also trained the classifier with a set of features that were derived without using any domain-specific feature or resources. The experiment is stated to achieve a precision value of 90.26%.

Nowadays, Twitter has become an open public platform to express opinions about political matters, government policies, economy and so on. A work from 2016 presents an approach to harness the political issue extractions and issue dependent standings and positions [10]. The authors developed the model to discover political issues and positions from unlabeled tweets. Their model is capable of discovering political issues and positions from an unlabeled dataset of tweets. The model estimates word-specific distributions (that denote political issues and positions) and hierarchical author/group-specific distributions (that show how these issues divide people). Their estimated distributions are then used to predict political affiliation with 68% accuracy.

Sikdar and Gambäck [11] demonstrates through a shared research task the experiment of classifying a large number of Twitter named entity system recognition, i.e., classifying a large number of twitter user's names, employing a supervised machine learning algorithm. The researchers classified the task into two distinguished parts such as, extracting the named entity from tweets in the first phase and classifying the names into ten different categorizations. The "Conditional Random Classifier" algorithm was trained on a feature-rich twitter dataset, and the obtained F1

score was 63.22%, while the F1 measure score based on the unseen test data was lower 40.06%.

A recent and one of the first research work on GST in India from 2017 demonstrates an approach for text mining and henceforth sentiment analysis of GST tweets [12]. The authors collected GST tweets during its implementation phase in India, developed a twitter data corpus. They implied the Naïve Bayes algorithm as the baseline algorithm of their work to gain polarity ratings of labeled tweets. After processing, with the help of tokens and cumulative tokens from GST specific tweets, the authors depicted the vicissitude of GST related buzzwords within the implementation phase, as well as the range, frequency, cumulative frequency and Zipf score of such most popular words. With that, the authors made a step to do a unified sentiment polarity percentage of the whole data corpus from the previously gained data insights.

Since, Twitter is a dynamic social platform to express our views and perspectives on the go, hence tweets often come out to be short, witty or satirical, rather than being utterly serious and long. Keeping this prospect in mind, the scope of satire detection in twitter and in other social network platforms is on the rise. Besides the traditional satire detection from mixed 'bag-of-words', the human hand-eye movement while reading such texts can also be taken into account for capturing the natural text processing flavor of human behaviors while reading something satirical or funny. Researchers have demonstrated their framework for managing this concept [13]. Apart from extracting the textual features, the researchers also emphasized the eye movement or gazing on texts while reading. They developed a CNN model, which can learn both from text features and gaze. For testing their model, they used an annotation of diverse people's reaction to reading the same text. With this bi-modal approach, the authors have established to show a better outcome for sarcastic texts. In a most recent work, the authors have proposed an "ontology" tool for sentiment analysis based on a large semantic network [14]. This tool not only helps to identify word sentiments, but it also produces the contexts, associated meaning with those words, and even their annotations linked with external resources. Instead of only following the keyword counts from social media texts, this work also utilizes the natural meaning of associated words that are being processed. Their proposed tool "OntoSenticNet" can detect expressed sentiments by analyzing the multiword expressions which are also related to other concepts that do.

Wang *et al* [15] used Long Short-Term Memory (LSTM) model for sentiment classification on twitter. Their system performed better than different classifiers which are based on feature engineering approaches. LSTM recurrent neural network processed negation expression phrase efficiently using multiplicative operations through gate structure compare to additives ones. Cambria [16] very nicely described the area of affective computing and sentiment

analysis in a different field such as sentiment and emotion analysis, recommendation and customer relationship management. They have divided the entire work into three main parts; (i) knowledge base method, (ii) statistical method and (iii) hybrid approach. Poria *et al* [17] first used 7-layer deep convolutional neural network for aspect identification in the area of opinion mining. They showed, for aspect extraction deep CNN is an effective approach than their discussed existing model. Salton *et al* [18] used attentive Recurrent Neural Network Language Models (RNN-LMs) model an extended model of (RNN-LM) for their task. They showed an attentive RNN-LM model's accuracy is better over the same dataset for taking less contextual information. Ma *et al* [19] used Sentic-long short-term memory (LSTM) is extended from LSTM for conducting their experiment. Sentic-LSTM model outperformed other state-of-the-art methods integrating target specific knowledge and commonsense knowledge.

While social media boasts a large number of users, which is also growing constantly, many bots are being used on social media for spreading malicious news, rumors, and hate statements and so on. Work has been done on detecting such types of bots on social media and to remove them on the basis of recall balance [20]. The researchers aimed to keep the precision rate high and obtained a balance between precision and recall to achieve the optimization results of removing the bots from social media.

Another work from 2016 shows the multidimensional polarity weightage for corpora based on the regional distributions [21]. Instead of taking the approach for conventional bipolarity analysis, i.e., positive and negative, the researchers took the step for multidimensional sentiment analysis on "valence arousal (VA) space", where a regional CNN-LSTM model can be deployed for inputting a text into several distinct regions, and from that, it can further be processed for extracting the information out of every regional CNN model. In such a scenario, the models of different regions can be able to produce heterogeneous information or feature for different regions. Based on that, it can also be categorized if the regional information has any long-distance dependency among them or not.

Ye *et al* [22] have added sentiment lexicon as an encoded manner into word vectors through a feedforward neural network with a CNN for their training purpose. Using this technique, they have got good accuracy over the standard sentiment analysis dataset.

Kenyon-Dean *et al* [23] have introduced a COMPLI-CATED class of sentiment to specify that sentiment does not belong only into positive and negative classes but also belongs to a COMPLICATED class. They have justified their logic-based two established dataset which are new twitter sentiment analysis (TSA) dataset, the McGill Twitter Sentiment Analysis dataset (MTSA).

Saleena [24] discussed the technique of ensemble classifiers where a single classifier has been formed by combining multiple base classifiers to improve the accuracy of

sentiment classification technique. For sentiment analysis, Diab and Hindi [25] have assigned proper weighted value in ensemble classification using multi-objective differential evolution. Symeonidis *et al* [26] used linguistic features, sentiment lexicon, and bag-of-words for combined supervised machine learning which is based on Majority voting scheming. To detect tweet polarity and analyzing opinions Azzouza *et al* [27] have used unsupervised machine learning techniques which helps to find out relevant keyword regarding the main topic of interest. They have developed a real-time system using the apache storm tool to track opinion on twitter. Tama and Rhee [28] used ensemble of weak classifiers approaches to envisage inactive students rather than a single classifier model on two real-world datasets. Omari and Al-Hajj [29] used machine and deep learning techniques for classification of Arabic language 34 articles of different domains. They used lexicon and corpus-based information for their work.

3. Preparing corpus on GST data

India has a number of 26.7 million active twitter users in 2017[5], i.e., currently, the second-highest in the world. Since GST was clearly one of the largest taxation reforms in the history of independent India, twitter witnessed a social opinion outburst on this topic mostly during Jun-July 2017, as it was the implementation phase of this tax reformation. In this context, we gathered tweets by employing live Twitter streaming API[6] in two major steps as follows.

At the early stage of our tweet streaming, we collected tweets in synchronization with the implementation phase of GST in India during June-July 2017. GST was implemented in the midnight of 30th June 2017 (i.e., 01.07.2017) on the presence of the members of both of the houses of the parliament of India. Naturally, GST became one of the top trending topics back then, and people were tweeting about it frequently rather than any other topics[7]. Since the evening time slot (between 6 p.m-10 p.m.) is considered to be the "prime-time" in India in terms of entertainment, news, debates and online social activities[8], we aimed primarily to stream the tweets between the aforesaid time window. While twitter API mostly allows its users to live-stream only 1-2% of the total tweets on any keywords, we observed that we were able to collect tweets at a rate of 24 thousand per day. This was also an indication of how this particular topic of choice was popular for tweeting in that time-phase. In this context, one thing is worth mentioning that up to October 2017, twitter used to support the highest of 140 characters per tweet including emojis and special characters, whereas, from 7th November 2017, twitter

---

[5] www.statista.com.

[6] https://developer.twitter.com.

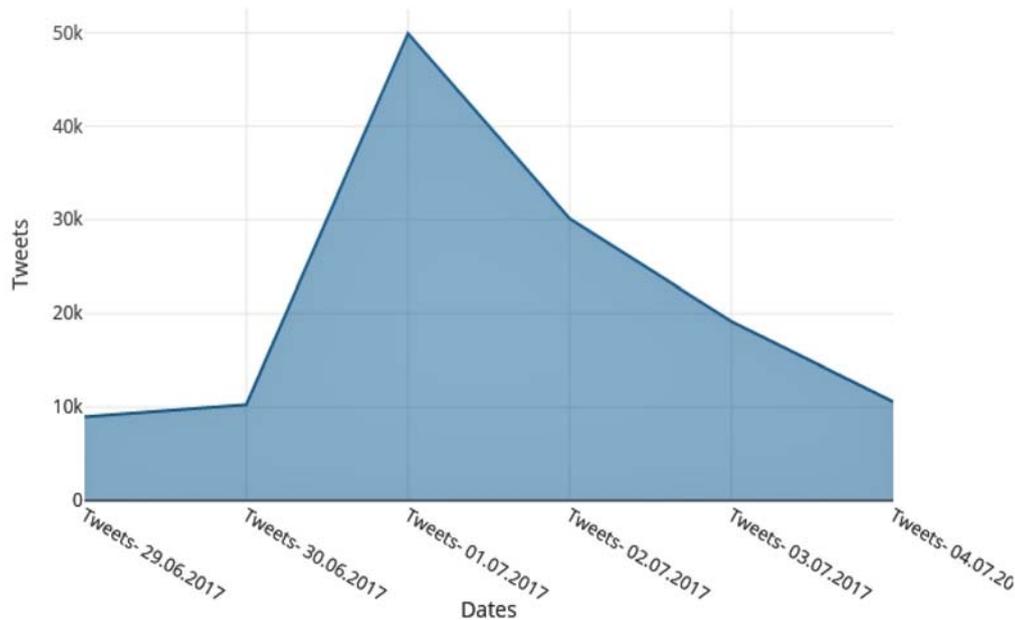[7] www.economictimes.indiatimes.com.

[8] www.business-standard.com.

Figure 1. Rise of GST tweets during Jun-July 2017.

expanded its character limits to 280 characters per tweet. However, since we have started collecting the GST tweets from June 2017, for most of the time (5 months out of 7 months), we were able to collect tweets with 140 characters only. One thing worth mentioning is that we are stating the total number of our collected tweets while taking all the impurities within tweets in the account. In figure 1, we represent the ascension and declination of GST tweets during its implementation week in India.

An initial visit on the collected data during this time period reveals several facts as observations, such as:

1) After the appliance of GST, the topic was settled within 2 or 3 months and it seemed that finally people were not tweeting about it like before,
2) While we managed to stream 24 k tweets per day during June-July 2017, in contrast we were only able to stream 3 k to 4 k tweets on GST later in September to October 2017.

Meanwhile, India's GST council held a meeting on 10th November 2017[9], for shifting rates of 177 products. This decision again slightly influenced the topic which motivated us again to collect the tweets as the second phase of our data collection. During this phase once again, we started collecting tweets at around 7 k to 8 k tweets per day.

Besides, as GST was already been implemented for few months at that time, our objectives were:

1) We also wanted to capture the opinions of the several ministers and ministries of govt. of India.
2) Also, we wanted to collect the criticisms against this tax or its effect on the economy in the form of tweets from the opponent political parties.

Hence, during this tweet collection phase, we streamed live tweets randomly both from the normal population as well as the twitter handles of *@narendramodi*, *@arunjaitley*, *@FinMinIndia*, *@RBI*, *@GST_Council*, *@RahulGandhi* and so on. After combining two phases with a span of almost 7 months, we gathered a number of 1,99,864 tweets, or almost 200 k unprocessed and raw tweets containing the hash-tagged keywords such as *#gst*, *#gsttax*, *#gstlaunch*, *#gstrollout*, *#gsteffect*, *#onenationonetax,* etc. among many other hash tags. along with the main tweet bodies. One of the main reasons behind choosing these particular hash tags was that we observed these were the most frequent hash tags from the very initial phase of our tweet collection.

4. Tweet modeling

While streaming tweets from twitter, a lot of heterogenous tweets based on solely different topics can get collected as long as they consist of the same themes of sentiment or same types of hash tagged words. To overcome this problem and to keep our tweet corpus as close to the GST topic as possible, we used a topic-sentiment model to stream the

---

[9]www.cbec.gov.in/resources.

live tweets. This model ensures the relevance of the tweets with GST tweets, as well as determines if the tweets contain any sentiment or not.

4.1 *Topic modelling*

In order to identify whether a tweet is relevant to our target/topic e.g., GST or not, we have considered a parameter κ as the keyword of the tweet. Now, a keyword makes the most impact determining the relevance of the tweet while streaming it from twitter. Moreover, as the tweeting person shifts the keyword position within the end of the tweet, i.e., closer towards the 140 characters limit, relevance of the keyword or its association with the tweet topic along with the context it is based on actually increases or decreases. More formally, if the keyword is found at the beginning of the tweet:

$$t_{posi} = \kappa + (n - \text{text}_j) \qquad (1)$$

where, t is the tweet itself, 'pos$_i$' is the position of the parameter, and here pos$_i$ = 1, n is the total tweet and text$_j$ is the remaining part of the tweet (j = n−1). Similarly, for equation (2), if the keyword is found in the middle of a tweet, the representation becomes:

$$t_{posi} = \frac{n - (\kappa - \text{text}_j)}{2} \qquad (2)$$

Finally, if the keyword is found at the last of a tweet:

$$t_{posi} = (n + \kappa) \qquad (3)$$

Now, combining all the possibilities of searching a relevant tweet for our target/topic, we formulated the model as in equation (4):

$$\sum_{i=0...n}^{pos_i} = \frac{na(\kappa + n)}{2} \qquad (4)$$

where t is the entire body of the tweet, α is the odd coefficient unit of keyword position and n is the remaining text position. Using this technique, the relevance of the tweet with GST and its associated words/phrases is achieved.

4.2 *Sentiment modeling*

After detection of the matching keyword(s) that we are looking for, the probability of finding the polarity from the remaining text as matching to our topic is determined and expressed as:

$$S_m = \kappa + s \qquad (5)$$

where S_m is the sentiment model, κ is the relevant keyword, and s is the sentiment expression found. The sentiment is of any flavor i.e., positive, negative or neutral. Only if the sentiment is found in a tweet along with the keyword, then

it can be streamed. If the tweet consists of both the keyword and sentiment, then only we streamed it. Finally, from equations (4) and (5), we form (6):

$$\sum_{i=0...n}^{pos_i} t = S_m \qquad (6)$$

We observed from the data that if we want to identify sentiments from tweets based on the GST as a target, not only the sentiment words directly linked with GST term, the other GST related words, which are Indian context specific (e.g., aadhar, demonetization, laws, etc.) can also contribute in identifying sentiments implicitly for tweets. Therefore, we have divided our tasks into two different subtasks; one is to identify sentiments for GST specific words based on state-of-the-art lexicons and another is to identify sentiments for GST related words based on the polarity-popularity model.

5. Feature I: GST-specified word sentiment identification

Streaming live tweets from twitter is a tricky task. Tweets generally comprise of spelling mistakes, unnecessary usage of string of alphabets within a short space, and SMS abbreviations (like LOL, LMAO, ROFL, BRB, BTW, etc.). We mostly aimed to remove Unicode, and URLs as they generally keep no impact on the extraction of underlying meaning or opinion of a natural English text. We also approached for removing the tweets with only GST relevant keywords and no tweet bodies, as those tweets do not bear any types of implacable information. At the same time, emoticons or emojis within the tweets were not eliminated, as emojis can be a useful key to determine the sentiment flavor of a text.

After preprocessing or cleaning the tweets, we tokenized the tweets into unigrams, bigrams, and trigrams with a frequency of 10000 words for each type. With that, we kept the frequency distribution of the words as *freq_dist(dense)*. The purpose behind this is to catch as many as unigrams, bigrams or trigrams possible within a tweet. With that, we also checked continuously for duplicate tokens when the tokens are being collected, until it reaches the EOF. Also, while extracting the grams, we removed the stop words from unigrams, but not from bigrams and tri-grams, as keeping the semantics of such phrases intact. After the stemming process of frequently occurring n-grams which we received previously, it helped to prevent the multiple occurrences of a single form of the word in many places of a document. Stemming was followed by part-of-speech (POS[10]) tagging and this helped to shrink down our filtered and extracted lexicons further.

---

[10]https://www.nltk.org/book/ch05.html.

5.1 *State-of-the-art lexicon based model*

After obtaining the final list of POS tagged words from our dataset, we intended to match them with five state-of-the-art sentiment lexicons, such as: *SenticNet 5.0* [30], *Vader* [31], *Positive_Negative Dataset* [32], *SentiWordNet 3.0* [33], and finally, *Twitter Sentiment Corpus* [34].

Our objective was to find out the coverage of our words in the standard lexicons. The coverage is presented in figure 2 as time vs. token growth graph. In figure 2, the *x-axis* represents an initial point where token matching started and the final point where all the tokens were matched with the aforesaid lexicons whereas the *y-axis* represents the number of tokens matched with time. We observed a linear growth of our lexicon when compared with the aforesaid standard sentiment lexicons and the newly matched tokens were listed in a separate file.

At first glance, it is obviously visible that our GST tweets could not find a large number of matching words with the previous state-of-the-art lexicons. In order to analyze the reason behind that, we observed as we already mentioned in section 1, most of the Indian twitter users do not tweet in thorough English language; rather they tend to use bilingual languages, or even complete native language with only the *#key-word* in English while tweeting. Thus, these tweets can be a mix of English-Bengali, English-Hindi, English-Punjabi, English-Tamil, or such other regional languages. However, for our approach, we have only streamed the tweets in proper English with relevant topics and sentiments; as stated before. We show a sample number of such code-mixed tweets in figure 3.

From the total number of matched words that we obtained, furthermore we approached for stemming and POS tagging of the words. These words, along with our previously extracted and POS tagged grams, further served as the mixed-bag-of-words for our popularity-polarity modelling.

6. GST-implied word identification

One of our key aims was to identify the words which occur repeatedly in our tweets and such words should be related with respect to the particular event of GST. However, they might not be found on any standard lexicon, or corpus before (e.g., aadhar, demonetization, etc.). Hence, in order to collect such important words that are specifically related to the event like GST and Indian circumstances, we used the distinct scores obtained from mixed-bag-of words approach based on Naïve Bayes. Furthermore, we made a file containing these words and their respective scores related to this type of economic event depending on a particular geopolitical region (in our experiment, for India). We adopted two parameters, namely Polarity and Popularity to identify the scores of such crucial words.

$$\text{Score(word)} \approx |Polarity| \quad \text{and}$$
$$\text{Score(word)} \approx |Popularity|$$

6.1 *Polarity-popularity model*

Where, polarity defines the sentiment rating from our previously stated sentiment score, and Popularity defines the number of occurrences for that word, as an instance, for 1,000 to 10,000 sample number of tweets from our entire dataset. Based on this equation, table 1 demonstrates the word polarity and popularity measure with respect to sentiment score and word occurrence.

Furthermore, if we denote word score as δ, hence for the changing value of δ, the topic score can be given as:

$$\delta = \text{Score(topic)} \approx \delta_1.|Polarity| + \delta_2.|Popularity| \quad (7)$$

Now, as we have already mentioned the compact relationship between the topic and sentiment in the tweets that we streamed, hence, the score of topic words is actually derived from the tweets consisting only GST topics and sentiments. More formally:

$$\sum_{i=0...n}^{pos_i} t = S_m = Score(topic) \quad (8)$$

This means that the polarity and popularity scores are derivable from each tweet with respective sentiment and topic.

Conclusively, the complete polarity and popularity model based on the topic-sentiment dependent tweet streaming can be expressed by (9):

$$\sum_{i=0...n}^{pos_i} t = S_m = \partial_1.|Polarity| + \partial_2.|Popularity| \quad (9)$$

where, δ_1 is sentiment polarity score, and δ_2 is the word occurrence count within a given number of tweets.

Applying the δ_1 (sentiment polarity score of any particular word) vs. δ_2 (occurrence of that word within 10,000 tweets) in a one to one combination, i.e., popularity vs. polarity calculation for our labelled and sentiment rated tweets, we found out a list of the most unique and only GST exclusive topic words within a shrink down number of 10,000 tweets. These words are frequently occurred words that appeared in the tweets that we crawled. The higher rating generated for any word, the higher it has appeared frequently within the tweets. We provided a combination of their popularity-polarity combinations. A few sample topic words from a total list of 9871 words, in alphabetical order along with their respective popularity score, are shown in table 2.

Once we obtained the complete list of words along with their respective popularity-probability scores, we calculated the polarity ratings of such occurred words. We made highest to lowest probability and normalized the ratings in a
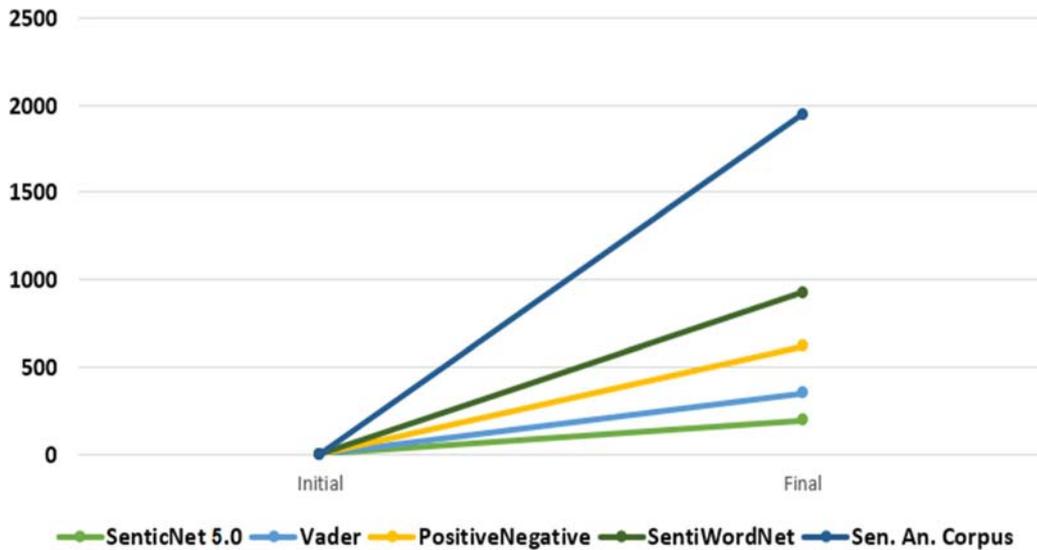
Figure 2. Coverage of word counts with state-of-the-art lexicons.

Figure 3. Tweets in completely regional Indian languages but with English keywords.

Table 1. Polarity and popularity measure of words with respect to sentiment score and word occurrence.

Popularity			*Polarity*		
δ_1	*Tweet Count*	Score (*word*) $\approx$ |*Polarity*|	δ_2	*Tweet Count*	Score (*word*) $\approx$ |*Popularity*|
1.0	1000	1000	556	1000	556000
2.0	2000	4000	500	2000	100000
3.0	4000	12000	353	4000	1412000
4.0	8000	32000	278	8000	2224000
5.0	10000	50000	130	10000	1300000

scale of 1 to 10 and plot a 3D word polarity cluster to classify them according to the polarity scores as in figure 4.

Utilizing the aforesaid sample number of tweets with their respective scores, we created a popularity-polarity model. It consists of a probability list containing the exclusive words related to the GST event, with their

respective probability score(s) to indicate the probability of their occurrence within a given range of tweets. This list further helped us to create the 3D word occurrence cluster and visualize the polarity and popularity graph.

From the figure, we observed that the words which are having a higher rating of more than 0 are positioned higher

Table 2. A few GST exclusive words are shown along with their respective probability of occurrence.

Word(s)	Popularity to Probability Score(s)
Aadhar	556.0
Allaboutgst	248.0
Demonitization	117.66
Economy	69.5
Govttax	26.0
Iitians	140.0
Jewellery	128.58
Kollywoodvoice	89.72
Laws	102.36

Polarity Cluster from Our GST Data Corpus
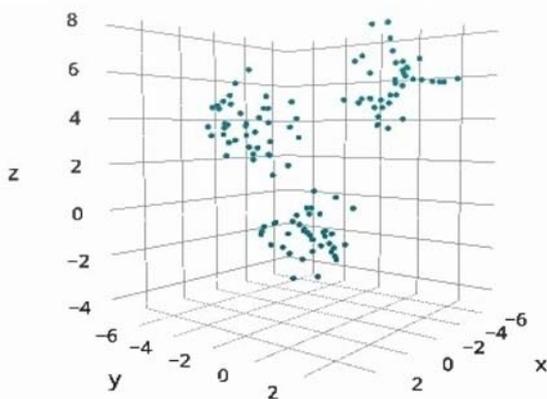
Figure 4. Word polarity cluster.

in the cluster, and considered as frequently occurred unique words. The data can be deployed for understanding the course and trend of such events beforehand. On the other side, words that are below 0 rates, are either common words, that appear with most of the trending topics, or they have very little impact on appearing again even if any such event takes place. From this list and data cluster, we also made a visual representation in figure 5 of the most occurred 36 words within just 1000 tweets. To observe the relations between popularity vs. the polarity of words, we plotted the following visual comparison in figure 6. This comparison graph represents the varieties of a sample number of words in the *x*-axis with their respective positive or negative threading parallels along with a minimized scale of polarity scores ranging from 0 *(very negative)* to 1 *(very positive)* in the *y*-axis, with also the intermediate polarity scales in between. Minimizing the polarity scores within a more condensed and simplistic range provided us the ability to produce a more compact visual representation of such words. We analyzed this relationship based on the

most popular words that we obtained previously. In this graph, the horizontal green bars are the indicators of carrying the 'positive' polarity tag for their respective words, while quite similar, the horizontal white threads in between depict the 'negative' polarity tagged words within a miniature group of (here a number of 1000) tweets. Besides visualizing this graph for a handful number of tweets, this analytical deduction can also be deployed for the entire data corpus.

7. Feature II: Sentiment rating of tweets

We retained the GST exclusive words from Feature I as a separate file. For assigning the sentiment rating, we developed an NLTK based Naïve Bayes sentiment analyzer to assign sentiment scores to the tweets which were previously labeled using the topic and sentiment to ensure their relevance with the particular subject matter (table 3). Since tweets are short in nature, for such short texts or textual fragments, Naïve Bayes tends to perform better than other baseline algorithms [35]. The scale of the sentiment scores was provided in the range from 1 to 5, such as *very negative (1.0)*, *negative (2.0)*, *neutral (3.0)*, *positive (4.0)*, and *very positive (5.0)*. Based on this scale and our classification, some real samples of the most appropriate tweets are shown in table 4 where each tweet belongs to exactly each sentiment label and sentiment rating.

8. LSTM for GST word sentiment prediction

We used the LSTM model [36] for sentiment prediction. Our model takes two different sets of inputs as Feature I and Feature II. As mentioned earlier, Feature I deals with the coverage of the extracted unigrams, bigrams and trigrams with some previous gold-standard datasets, which eventually helped us to build the popularity-polarity model. This model generates a large number of GST exclusive words, specifically 9871 words. We then convert these words using *word2vec* and fed the vectors batch by batch as the first set of inputs. On the other hand, Feature II contains 80% of the tweets from our GST corpus, which comprise sentiment rating varying a range from *most negative* (1.0) to *most positive* (5.0) and other polarity scores in between. These tweets are converted from dictionary to vector values, using *doc2vec*, and these vectors are fed batch by batch as the second set of inputs.

For Feature II, we split the ratio of our tweets 80:20 to train and then to test the sentiment prediction of the tweets. 80% of training data is labelled and sentiment rated, while the rest of the 20% test data is also preprocessed, labelled and non-sentiment rated data. Hence the sentiment prediction outcome on this test data determines the success of the word generation using polarity vs. popularity model and
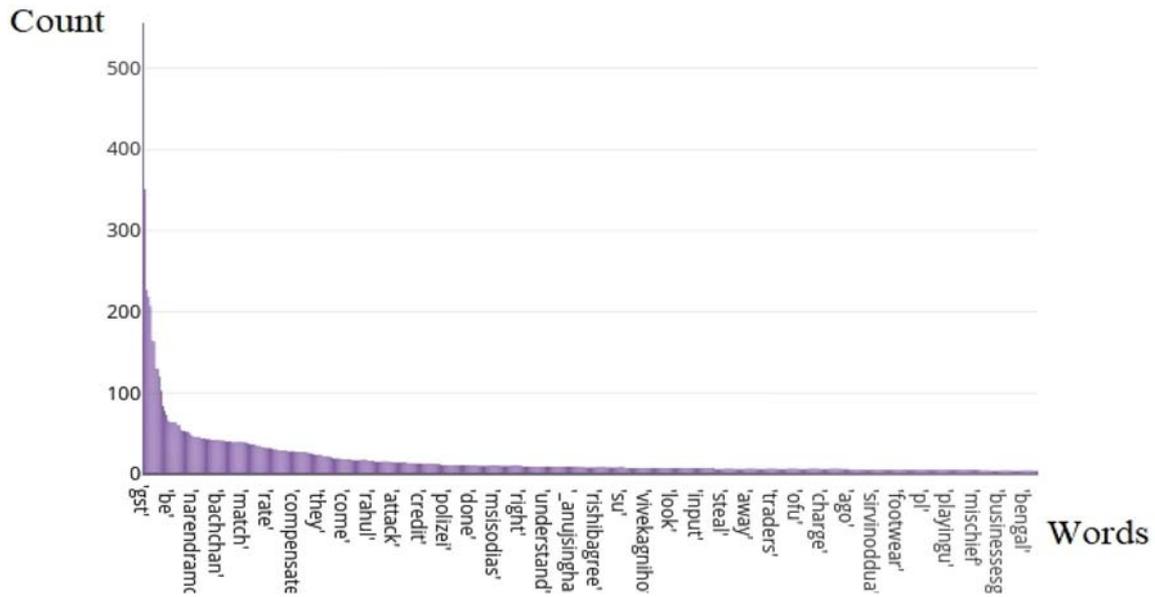
Figure 5. Word popularity count within a sample number of 1,000 tweets.
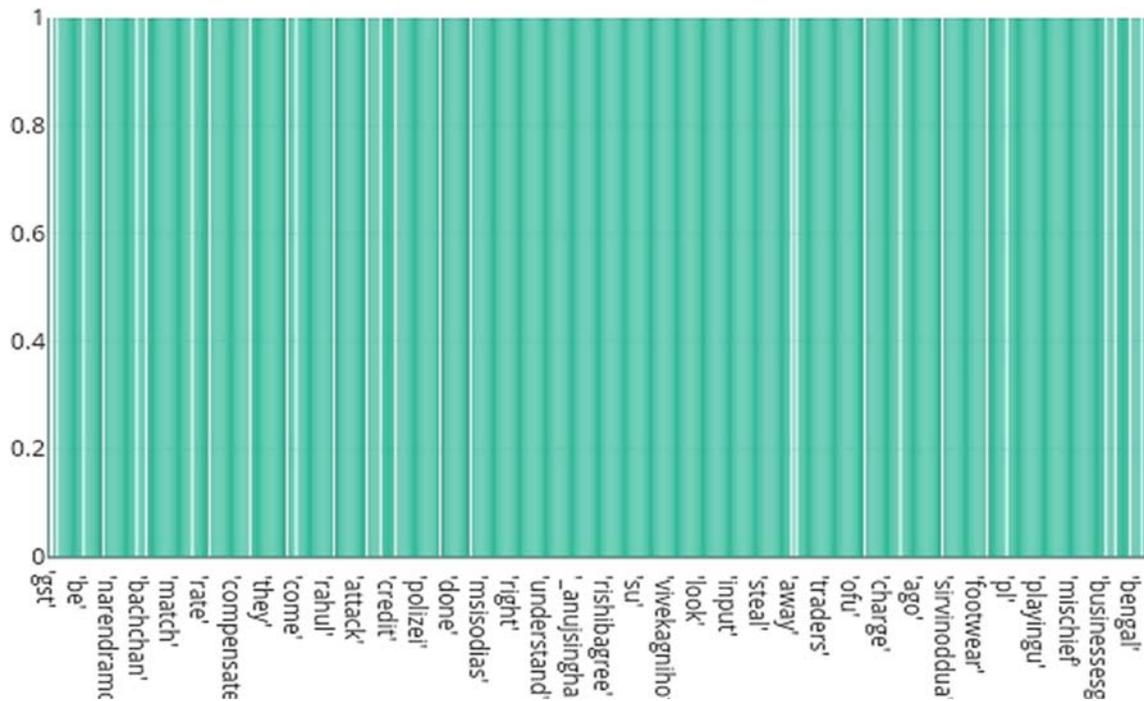

Figure 6. Polarity mapping for most frequent GST exclusive words.

sentiment rating thus far. In the training process of the LSTM model, *x_train* method is used to split the training and testing branches and *x_words* method is used to hold the vectors into a temporary memory location and employing them for training per batch.

Our LSTM model capable of converting *word2vec* and *doc2vec* from mixed-bag-of-words and sentiment rated datasets with natural English tweets, and after training, it successfully evaluates the sentiment predictions. In our experiment, we added a sequential and dense LSTM model using the *Tensorflow*[11] framework and importing *Keras*[12]

---

[11]https://www.tensorflow.org/.

[12]https://keras.io/layers/recurrent/#lstm.

Table 3. Example of tweets belonging to each Sentiment Class.

Index	Tweet	Class	Polarity Rating
1	"@CNBCTVLive These anchors don't know actual rules of GST how talk about that I don't know !!"	Very Negative	1.0
2	"I think #GST will force us to go back to the ages where bear trade was practiced."	Negative	2.0
3	"@INCIndia Watch Press briefing by State Ministers Manpreet Badal, Kamala Kaan @krishnabgowda on #GST"	Neutral	3.0
4	"@timesofindia #GST gets simpler for small businesses and exporters."	Positive	4.0
5	"@jitu_vaghani Congratulate PM narendramodi ji and FM arunjaitley ji on taking landmark decisions and making #GST even simpler."	Very Positive	5.0

Table 4. Comparative performance analysis among different activation functions of LSTM.

Activation Function(s), Epoch@10,000	Loss	Accuracy
LSTM w. *sigmoid*	47.06%	84.51%
LSTM w. *hard_sigmoid*	52.34%	79.57%
LSTM w. *softmax*	68.21%	61.35%
LSTM w. *linear*	73.17%	55.51%
LSTM w. *relu*	78.62%	50.68%
LSTM w. *tanh*	81.28%	49.65%

library. Based on our exclusive words set, we set the vector dimension of each record to 200 and a batch size to four elements from each record named as *index*, *tweet*, *sentiment class*, and *sentiment rating*. Our model has six layers and 1000 paddings per batch. We represent a miniature diagrammatic version of our LSTM model architecture in figure 7. For a comparative analysis between the activation functions to find out which one is the best suited for our approach, we incorporated a total of 6 activation functions for measuring the performance parameters of the LSTM model, since these activation functions are generally used most in textual analysis tasks [37, 38]. These activation functions are *hard_sigmoid, sigmoid, linear, relu, softmax* and *tanh*. Our model is trained for a total of 60,000 epochs as 10,000 epochs for each six activations functions using parameters *optimizer, loss and accuracy*.

Table 1 shows the accuracy of our model. We fixed the epoch for training as 10,000 and finally, we compiled our model using parameters like an *optimizer, loss,* and *accuracy* and observed that after completing 10,000 epochs for each of 6 activation functions, i.e., a total of 60,000 epochs, the following accuracies were achieved as shown in table 4.

From the table, we can see that we achieved an accuracy of 84.51% with our developed LSTM model using the *sigmoid* activation function, which is the highest among all the distinct several functions that we used. To validate the input features of the model as well as results produced, we discuss them in subsequent sections.

9. Experimental results

Simultaneously with sentiment prediction, our LSTM model generated a file containing the tweets that it predicted as *positive, negative* and *neutral*. We compared this file with our already standardized sentiment rated corpus to validate the prediction-based analysis. We compared the predicted tweets with the actual tweets to find out the *confusion matrix* consisting of a *true positive, true negative, false positive* and *false negative*. Using these features, we further calculated *precision, recall, accuracy,* and finally *f1 score*. For the analysis, we used a shrink down a fractional sample of the whole data corpus, i.e., for a number of 10,000 tweets from our entire dataset. The main reason behind this is to reduce the time complexity as much as possible, by making the overall analysis much faster. Another reason is that a fractional overview of the results can be the showcase of the entire data corpus's characteristics.

This analytical observation of our data corpus also represents the error rate in both respects to positive predicted and negative predicted tweets. With 10,000 tweets, our training (standard) set has total of 9908 entries (tweets) with 5413 actual positive (54.63% of 9908), 4100 actual negative (41.38% of 9908) and 395 actual neutral (25.08% of 9908) tweets. Our validation set has a number of 4919 positive predicted tweets (90.83%), 3468 negative predicted tweets (84.58%) and 531 neutral predicted tweets (74.38%). We represent the predicted results in table 5.

Next, we present the classification report in table 6, in which we show the statistical measures calculated from table 6. Entries with 43.01% negative, 56.99% positive and test set has total 5413 entries with 41.94% negative, 58.06% positive with an overall accuracy score of 83.44%. The validation result for 10,000 tweets reveals that the null accuracy is 54.72% whereas the overall accuracy score is 83.87%, which is 29.13% more accurate than null accuracy.

Next, we did a comparison of our results in table 7 with four already established state-of-the-art sentiment lexicons: *Linguistic Inquiry Word Count (LIWC)* [39], *General*
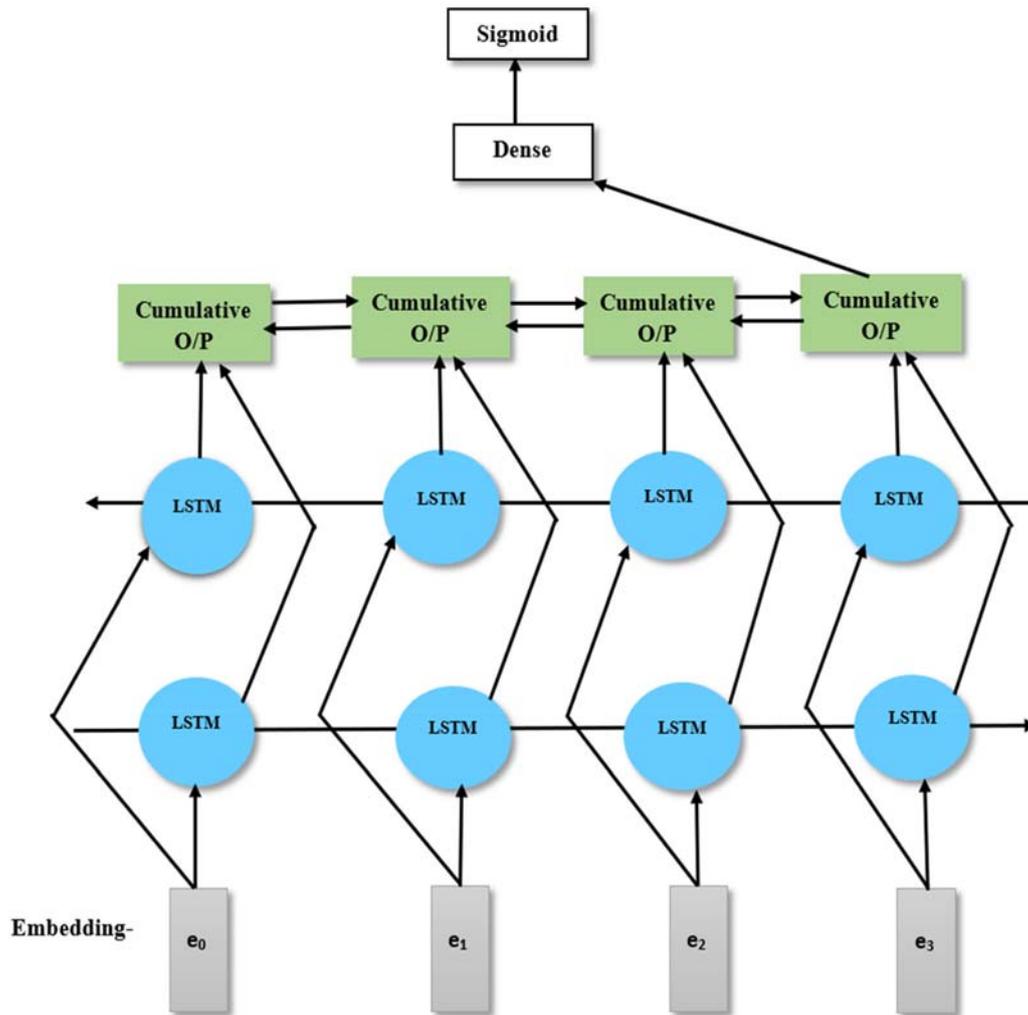
Figure 7. A representative diagram of our LSTM model.

Table 5. Confusion Matrix & Classification Report shows the differences between the predicted and actual tweets along with the prediction parameters.

Actual		Predicted	
		Positive	*Negative*
	Positive	4919 (TP)	497 (FP)
	Negative	634 (FN)	3468 (TN)

Table 6. Confusion Matrix & Classification Report shows the differences between the predicted and actual tweets along with the prediction parameters.

Precision	Recall	Accuracy	F1 Score
0.9039	0.8863	0.8387	0.899

Inquirer (GI) [40], *Affective Norms for English Words (ANEW)* [41], *Word-Sense Disambiguation (WSD)* using *WordNet* [42], for analyzing the performance of our sentiment-based accuracy along with the classification parameters. The previous works are mainly based on a cumulation of manual human ratings on some particular textual topics from time to time. As shown in table 7, in most scenarios,

our approach outperforms the other previously well-established lexicons for sentiment analysis. In the case of social media texts (here tweets), our approach provides better overall classification parameters than the manually given human ratings in the previous experimental works.

Furthermore, table 8 represents the comparative analysis of classification report of our model on the New York Times annotated corpus[13] benchmark dataset and

---

[13]https://dss.princeton.edu/catalog/resource1357.

Table 7. Comparison of classification performance on social media posts.

	Overall Accuracy ground truth (of the respective corpora)	Classification Accuracy Metrics		
		Overall Precision	Overall Recall	Overall F1 Score
Social Media Texts (Range: 10,000 Tweets)				
Our Approach	0.838	0.90	0.88	0.89
GI [39]	0.512	0.79	0.51	0.67
LIWC [40]	0.606	0.91	0.49	0.60
ANEW [41]	0.451	0.79	0.46	0.57
WSD [42]	0.401	0.69	0.44	0.52

Table 8. Comparison of classification performance on NY Times Editorials.

	Classification Accuracy Metrics		
Methods	Overall Precision	Overall Recall	Overall F1 Score
Context-labelled Texts (NY Times 3,500 article review)			
Ours	0.81	0.78	0.72
Joshi *et al* [43]	0.42	0.51	0.55
Bouazizi and Ohtsuki [44]	0.54	0.49	0.50
Ghosh *et al* [45]	0.30	0.50	0.42
Amir *et al* [46]	0.44	0.48	0.57
Abercrombie and Hovy [47]	0.39	0.51	0.51
Joshi *et al* [48]	0.45	0.21	0.23
Mishra *et al* [49]	0.35	0.41	0.40
Muresan *et al* [50]	0.45	0.48	0.54
Bouazizi and Ohtsuki [51]	0.38	0.37	0.55
Fersini *et al* [52]	0.41	0.41	0.36
Bharti *et al* [53]	0.47	0.58	0.49
Liu *et al* [54]	0.52	0.54	0.60
Wang *et al* [55]	0.55	0.54	0.57
Hernández-Farías *et al* [56]	0.70	0.59	0.52

newspaper editorials against some of the state-of-the-art experiments.

9.1 *Error analysis*

In our LSTM model-based experiment, we used exclusively generated words from popularity vs. polarity model, and sentiment rated tweets in each epoch as input for converting *word2vec* and *doc2vec* and building the library for training and thereafter testing. To find accuracy and loss of each 1000th epochs run on the LSTM model with a sigmoid activation function.

For we employed our LSTM model for 10,000 epochs, we evaluated our model with sigmoid activation function for each of the 1000th epoch, for analyzing the accuracy and loss. We plotted the accuracy vs. loss graph in figure 8 to evaluate our model's performance and analyze the errors further.

The above report shows that we could not achieve more than 84.51% due to the loss rate of 46-47%. Since our loss rate was 46-47% for most of the time, and the accuracy did not reach beyond 84.51%. While analyzing the probable reasons behind this, we have tried with 1,30,000 clean pre-processed tweets, and we evaluated our LSTM model for 10,000 epochs, hence we can say the model took 13,000 tweets in each epoch as input for converting *word2vec* and building the library for training and thereafter testing.

In our next step, we split the train and test ratio as 80:20, hence the first 8 epochs were implied for training and the last 2 epochs, i.e., 9,000[th] and 10,000[th] epoch is actually responsible for predicting the accuracy. Thus, 13,000 tweets are approximately fed as input in each of the epochs. We further had a closer observation of our model, as well as on these tweets.

We found out that even keeping 84.17% as the average accuracy threshold, 2000th, 5000th, and 6000th epoch did not produce an overall improvement inaccuracy. While
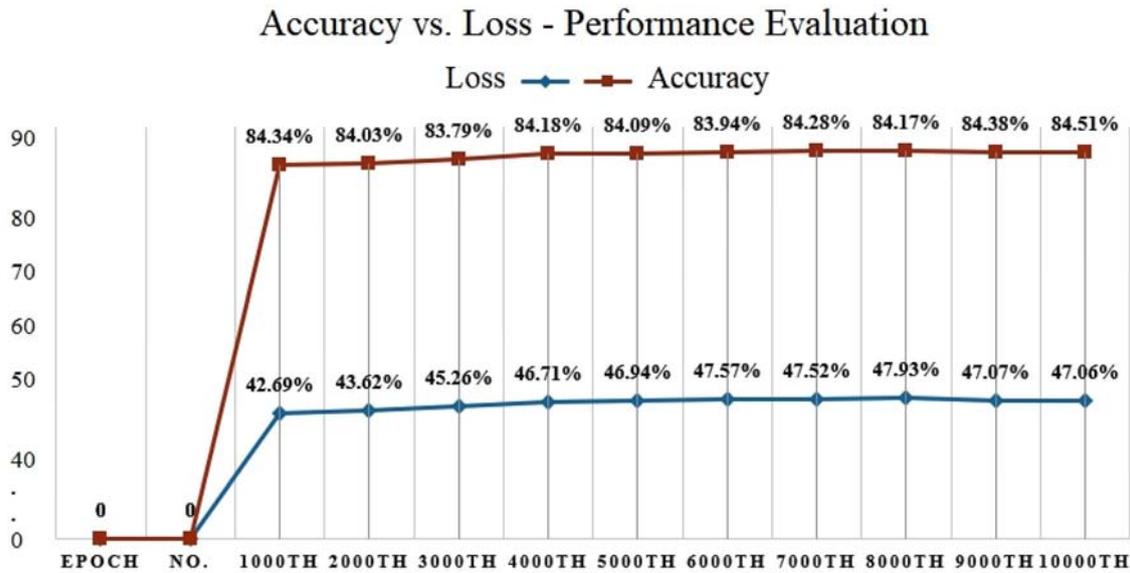
Figure 8. Performance evaluation graph (Accuracy vs. Loss) of our LSTM model.

analyzing the reason of performance drop in these three epochs, we find the following reasons.

(1) These samples of tweets are heavily code-mixed; our model could not achieve predicting testing accuracy.
(2) Since we are not working with the bi-lingual tweet and language-mixed tweets, now we are not able to address the problem, in this scope, we would further like to address this problem as an extension of this work.

In table 9, we take a look at the sample type of tweets among other hundreds of such tweets that were fed as inputs in these 3 epochs. These are only some samples from the tweets which are heavily code-mixed. This is one area where our model could not hit the bulls-eye while predicting the testing accuracy. Since we are not working with bi-lingual tweets and language-mixed tweets, in this scope, we would further like to address this problem as an extension of this work.

10. Conclusion and future work

In this study, we represented a deep learning inspired lexical-level sentiment analysis of GST tweets. We approached for a topic-sentiment based tweet crawling and thereafter word polarity vs. popularity generation for discovering and clustering the GST exclusive topic words from GST tweets in India. We collected tweets on GST for a course of 7 months, pre-processed and filtered the tweets, extracted unigrams, bi-grams and trigrams from the tweets, compared these grams with previous state-of-the-art lexical and twitter datasets. We separated the words that we found are matching and did the stemming and POS tagging. This helped us to create a Bag-of-Words from the GST tweets which we retained separately. Simultaneously, we developed an NLTK based Naïve Bayes sentiment analyzer; we gave our twitter dataset sentiment ratings on a scale of 1.0 to 5.0. Now using the bag-of-words that we obtained, and the sentiment rated tweets, we developed a sentiment-trend model, by which we were able to generate the scores for

Table 9. Tweets which caused the probable performance drop.

Sample Tweet	Input In	Average Prediction Accuracy (%)	Actual Prediction Accuracy (%)
"@LMKMovieManiacathuepdi #GST vanthumrenduthadava poi pakkurangalo….ungalukuenaosi ticket dana ??????"	2000th epoch	84.17%	84.03%
"@MrsGandhi J&K oppositnldrskokyataklif? Rich enuf to last 3 generations. Coming in way of dvlpmt for all #kashmiris by opposing #OneNationOneTax"	5000th epoch	84.17%	84.09%
"@KyaUkhaadLega: "Food items sastezaroorhogayehain par tum control rakhna, aurbhi log haindeshmei?" #GST"	6000th epoch	84.17%	83.94%

word popularity and polarity for most occurred words in the GST related tweets during the course of GST implementation phase in India. We identified a number of 9871 words within a miniature sample of 10,000 tweets from our entire data corpus and visualized their sentiment polarity using a 3D data cluster and polarity vs. polarity mapping as a whole. Now using this newly developed rated dataset, we implied our LSTM model for training and testing of our data. We kept a split of 80:20, i.e., 80% data for training, and rest 20% data for testing. After 10,000 epochs we achieved an accuracy of 84.51%.

However, for our future work, we want to keep the most occurred words from this event, and we want to deploy them if any such event takes place, to predict the course and trend of that event. With that, we would also like to develop a system to successfully evaluate bi-lingual font-mixed tweets to enhance the accuracy of our experiment. Finally, since our work is one of the first works on such a large-scale economic reform, we are keen to publish our GST data with all the components on an open-source data repository platform like Github[14] in near future, so that the other researchers feel free to experiment with our findings from this event, and they can compare their achieved results with that of ours.

References

[1] Mandal S, Mahata S and Das D 2018. Preparing Bengali-English Code-Mixed Corpus for Sentiment Analysis of Indian Languages. In: *ALR, Workshop on Asian Language Resources, LREC*, pp. 57–64

[2] Bartusiak R, Augustyniak Ł, Kajdanowicz T, Kazienko P and Piasecki M 2019. Wordnet2vec: Corpora agnostic word vectorization method. *Neurocomputing 326*: 141–150

[3] Tang D, Qin B and Liu T 2015. Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of the 2015 conference on empirical methods in natural language processing* pp. 1422–1432

[4] Saha S and Ekbal A 2013. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering 85*: 15–39

[5] Chawla K, Ramteke A and Bhattacharyya P 2013, June. Iitb-sentiment-analysts: Participation in sentiment analysis in twitter semeval 2013 task. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* pp. 495–500

[6] Balamurali AR, Khapra MM and Bhattacharyya P 2013, March. Lost in translation: viability of machine translation for cross language sentiment analysis. In: *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 38–49). Springer, Berlin, Heidelberg

[7] Wang H, Can D, Kazemzadeh A, Bar F and Narayanan S 2012, July. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In: *Proceedings of the ACL 2012 system demonstrations* pp. 115–120. Association for Computational Linguistics

[8] Khattri A, Joshi A, Bhattacharyya P and Carman M 2015, September. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In: *Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis* pp. 25–30

[9] Akhtar MS, Sikdar UK and Ekbal A 2015, July. IITP: Hybrid approach for text normalization in Twitter. In: *Proceedings of the Workshop on Noisy User-generated Text* pp. 106–110

[10] Joshi A, Bhattacharyya P and Carman M 2016, June. Political issue extraction model: A novel hierarchical topic model that uses tweets by political and non-political authors. In: *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* pp. 82–90

[11] Sikdar UK and Gambäck B 2016, December. Feature-rich twitter named entity recognition and classification. In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)* pp. 164–170

[12] Das S and Kolya AK 2017, November. Sense GST: Text mining & sentiment analysis of GST tweets by Naive Bayes algorithm. In: *2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)* pp. 239–244. IEEE

[13] Mishra A, Dey K and Bhattacharyya P 2017, July. Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pp. 377–387

[14] Dragoni M, Poria S and Cambria E 2018. OntoSenticNet: A commonsense ontology for sentiment analysis. *IEEE Intelligent Systems 33*(3): 77–85

[15] Wang X, Liu Y, Sun CJ, Wang B and Wang X 2015, July. Predicting polarities of tweets by composing word embeddings with long short-term memory. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* pp. 1343–1353

[16] Cambria E 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems 31*(2): 102–107

[17] Poria S, Cambria E and Gelbukh A 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems 108*: 42–49

[18] Salton G, Ross R and Kelleher J 2017, November. Attentive language models. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* pp. 441–450

[19] Ma Y, Peng H and Cambria E 2018, April. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: *Thirty-second AAAI conference on artificial intelligence*

[20] Morstatter F, Wu L, Nazer TH, Carley KM and Liu H 2016, August. A new approach to bot detection: striking the balance between precision and recall. In: *2016 IEEE/ACM*

International Conference on Advances in Social Networks Analysis and Mining (ASONAM) pp. 533–540. IEEE

[21] Teng Z, Vo DT and Zhang Y 2016, November. Context-sensitive lexicon features for neural sentiment analysis. In: *Proceedings of the 2016 conference on empirical methods in natural language processing* pp. 1629–1638

[22] Ye Z, LiF and Baldwin T 2018, August. Encoding sentiment information into word vectors for sentiment analysis. In: *Proceedings of the 27th International Conference on Computational Linguistics* pp. 997–1007

[23] Kenyon-Dean K, Ahmed E, Fujimoto S, Georges-Filteau J, Glasz C, Kaur B, Lalande A, Bhanderi S, Belfer R, Kanagasabai N and Sarrazingendron R 2018, June. Sentiment Analysis: It's Complicated! In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* pp. 1886–1895

[24] Saleena N 2018. An ensemble classification system for twitter sentiment analysis. *Procedia Computer Science* 132: 937–946

[25] Diab DM and El Hindi KM 2017. Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification. *Applied Soft Computing* 54: 183–199

[26] Symeonidis S, Effrosynidis D, Kordonis J and Arampatzis A 2017, August. DUTH at SemEval-2017 Task 4: a voting classification approach for Twitter sentiment analysis. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* pp. 704–708

[27] Azzouza N, Akli-Astouati K, Oussalah A and Bachir SA 2017, June. A real-time Twitter sentiment analysis using an unsupervised method. In: *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics* pp. 1–10

[28] Tama BA and Rhee KH 2018. A comparative study of classifier ensembles for detecting inactive learner in university. *International Journal of Data Analysis Techniques and Strategies* 10(4): 351–368

[29] Omari MA and Al-Hajj M 2020. Classifiers for Arabic NLP: survey. *International Journal of Computational Complexity and Intelligent Algorithms*, 1(3): 231–258

[30] Cambria E, Poria S, Hazarika D and Kwok K 2018, April. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In: *Thirty-Second AAAI Conference on Artificial Intelligence*

[31] Hutto CJ and Gilbert E 2014, May. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: *Eighth international AAAI conference on weblogs and social media*

[32] Hu M and Liu B 2004, August. Mining and summarizing customer reviews. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 168–177

[33] Baccianella S, Esuli A and Sebastiani F 2010, May. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: *Lrec* Vol. 10, No. 2010, pp. 2200–2204

[34] Sanders NJ 2011. Sanders-twitter sentiment corpus. *Sanders Analytics LLC*, 242

[35] Wang S and Manning CD 2012, July. Baselines and bigrams: Simple, good sentiment and topic classification. In:

Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2 pp. 90–94. Association for Computational Linguistics

[36] Dai AM and Le QV 2015. Semi-supervised sequence learning. In: *Advances in neural information processing systems* pp. 3079–3087

[37] Glorot X, Bordes A and Bengio Y 2011, June. Deep sparse rectifier neural networks. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics* pp. 315–323

[38] Severyn A and Moschitti A 2015, August. Twitter sentiment analysis with deep convolutional neural networks. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* pp. 959–962

[39] Pennebaker JW, Boyd RL, Jordan K and Blackburn K 2015. *The development and psychometric properties of LIWC2015*.

[40] Wilson T, Wiebe J and Hoffmann P 2005, October. Recognizing contextual polarity in phrase-level sentiment analysis. In: *Proceedings of human language technology conference and conference on empirical methods in natural language processing* pp. 347–354

[41] Bradley MM and Lang PJ 1999. *Affective norms for English words (ANEW): Instruction manual and affective ratings* (Vol. 30, No. 1, pp. 25–36). Technical report C-1, the center for research in psychophysiology, University of Florida

[42] Gao N, Zuo W, Dai Y and Lv W 2014. Word sense disambiguation using wordnet semantic knowledge. In *Knowledge Engineering and Management* Springer, Berlin, Heidelberg. pp. 147–156

[43] Joshi A, Tripathi V, Patel K, Bhattacharyya P and Carman M 2016. Are word embedding-based features useful for sarcasm detection?. *arXiv preprint* arXiv:1610.00883.

[44] Bouazizi M and Ohtsuki T 2016, May. Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in Twitter. In: *2016 IEEE International Conference on Communications (ICC)* pp. 1–6. IEEE.

[45] Muresan S, Gonzalez-Ibanez R, Ghosh D and Wacholder N 2016. Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology* 67(11): 2725–2737

[46] Amir S, Wallace BC, Lyu H and Silva PCMJ 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint* arXiv:1607.00976

[47] Abercrombie G and Hovy D 2016, August. Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. In: *Proceedings of the ACL 2016 Student Research Workshop* pp. 107–113

[48] Joshi A, Tripathi V, Bhattacharyya P and Carman M 2016, August. Harnessing sequence labeling for sarcasm detection in dialogue from tv series 'friends'. In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* pp. 146–155

[49] Mishra A, Kanojia D, Nagar S, Dey K and Bhattacharyya P 2017. Harnessing cognitive features for sarcasm detection. *arXiv preprint* arXiv:1701.05574

[50] Musi E, Ghosh D and Muresan S 2016, August. Towards feasible guidelines for the annotation of argument schemes.

In: *Proceedings of the third workshop on argument mining (ArgMining2016)* pp. 82–93

[51] Bouazizi M and Ohtsuki T 2015, August. Opinion mining in twitter how to make use of sarcasm to enhance sentiment analysis. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015* pp. 1594–1597

[52] Fersini E, Pozzi FA and Messina E 2015, October. Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers. In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* pp. 1–8. IEEE

[53] Bharti SK, Babu KS and Jena SK 2015, August. Parsing-based sarcasm sentiment recognition in twitter data. In: *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* pp. 1373–1380. IEEE

[54] Liu P, Chen W, Ou G, Wang T, Yang D and Lei K 2014, June. Sarcasm detection in social media based on imbalanced classification. In: *International Conference on Web-Age Information Management,* Springer, Cham. pp 459–471

[55] Wang Z, Wu Z, Wang R and Ren Y 2015, November. Twitter sarcasm detection exploiting a context-based model. In: *International Conference on Web Information Systems Engineering* . Springer, Cham. pp. 77–91

[56] Hernández-Farías I, Benedí JM and Rosso P 2015, June. Applying basic features from sentiment analysis for automatic irony detection. In: *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, Cham. pp. 337–344